Atty Docket No. 50291.P009 *Patent*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:	
John R. Koza et al.	Examiner: Wong, Lut
Application No.: 09/548,637	Art Unit: 2129
Filed: April 13, 2000	Confirmation No.: 6771
For: A METHOD AND APPARATUS FOR DESIGNING STRUCTURES	

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

This is an appeal to the Board of Patent Appeals and Interferences from the decision of the Examiner of Group 2129, mailed January 12, 2011, in which claims 1-6, 13, and 15-23 in the above-identified application were rejected in a final action. This Appeal Brief is hereby submitted pursuant to 37 C.F.R. § 41.37(a).

CERTIFICATE OF TRANSMISSION
I hereby certify this correspondence is being submitted via EFS-Web to the United States Patent and Trademark Office on July 12, 2011.
/Andreas J. Radke/
Andreas J. Radke

I. REAL PARTY IN INTEREST

The real party in interest is the assignee of the full interest in the invention, Dr. John R. Koza, 25372 La Rena Lane, Los Altos Hills, California 94022.

II. RELATED APPEALS AND INTERFERENCES

To the best of Appellant's knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision in the instant appeal.

III. STATUS OF THE CLAIMS

Claims 1-6, 13, and 15-23 are pending in the application and were finally rejected in an Office Action mailed January 12, 2011. Claims 1-6, 13, and 15-23 are the subject of this appeal. A copy of Claims 1-6, 13, and 15-23 as they stand on appeal are set forth in Appendix A.

IV. STATUS OF AMENDMENTS

No amendments have been submitted subsequent to the Final Office Action mailed January 12, 2011.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Appellant's invention as claimed in claims 1-6, 13, and 15-23 is directed to a method and apparatus for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure, using genetic programming techniques.

Independent claim 1 claims a method including: initializing a plurality of candidate entities and an iteration count with a predetermined value by supplying, from an external source, at least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities (Figs. 1A-1B and 2; line 12 of page 34 – line 7 of page 35, line 20 of page 41 – line 4 of page 42), wherein each candidate entity is represented by a tree structure having a plurality of nodes representing a structure of the candidate entity (line 7 of page 36 – line 2 of page 37).

Independent claim 1 further performs iterative genetic programming operations, each iteration including: creating a description of the structure for each of the candidate entities based on its tree structure (Figs. 1A-2B and 2; lines 8-13 of page 35, lines 17-19 of page 42), analyzing behavior and characteristics based on the description of the structure of each candidate entity, including a simulation of the structure (Figs. 1A-1B and 2; lines 3-15 of page 37, lines 20-22 of page 42), comparing each of the plurality of candidate entities with

the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity (Figs. 1A-1B and 2; lines 6-12 of page 38, lines 1-3 of page 43), the isomorphism value representing a dissimilarity between the respective candidate entity and the reference structure (lines 4-6 of page 89 and line 1 of page 94 to line 7 of page 95), determining a fitness value for each of the candidate entities based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity (Figs. 1A-1B and 2; lines 8-20 of page 35, lines 13-22 of page 38, lines 4-7 of page 43), selecting at least one candidate entity from the plurality of candidate entities that has a fitness value exceeds a predetermined threshold (Figs. 1A-1B and 2; lines 8-15 of page 40), creating at least one new candidate entity by creating a variation in the selected at least one candidate entity if the selected at least one candidate does not satisfy the predetermined design requirement or a number of iterations has not reached the predetermined value of the iteration count, including performing one of a reproduction operation, offspring crossover operation, mutation operation, and an architecture altering operation on the at least one selected candidate entity (Figs. 1A-1B and 2; line 21 of page 35 to line 6 of page 36, line 7 of page 39 to line 12 of page 41, lines 8-14 of page 43), and terminating the iterations if the selected at least one candidate satisfies the predetermined design requirement or a number of iterations has reached the predetermined value of the iteration count, wherein at least one of the selected candidate entities is used to design an end-result structure in view of the predetermined design requirement, wherein the end-result structure does not possess key characteristics of the reference structure; and updating the iteration count at the end of each iteration (Figs. 1A-1B and 2; line 17 of page 34 to line 7 of page 35, lines 5-16 of page 42).

Independent claim 22 claims a computer system to perform a process of creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure. The process includes: initializing a plurality of candidate entities and an iteration count with a predetermined value by supplying, from an external source, at

least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities (Figs. 1A-1B and 2; line 12 of page 34 – line 7 of page 35, line 20 of page 41 – line 4 of page 42), wherein each candidate entity is represented by a tree structure having a plurality of nodes representing a structure of the candidate entity (line 7 of page 36 – line 2 of page 37).

Independent claim 22 further performs iterative genetic programming operations, each iteration including: creating a description of the structure for each of the candidate entities based on its tree structure (Figs. 1A-2B and 2; lines 8-13 of page 35, lines 17-19 of page 42), analyzing behavior and characteristics based on the description of the structure of each candidate entity, including a simulation of the structure (Figs. 1A-1B and 2; lines 3-15 of page 37, lines 20-22 of page 42), comparing each of the plurality of candidate entities with the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity (Figs. 1A-1B and 2; lines 6-12 of page 38, lines 1-3 of page 43), the isomorphism value representing a dissimilarity between the respective candidate entity and the reference structure (lines 4-6 of page 89 and line 1 of page 94 to line 7 of page 95), determining a fitness value for each of the candidate entities based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity (Figs. 1A-1B and 2; lines 8-20 of page 35, lines 13-22 of page 38, lines 4-7 of page 43), selecting at least one candidate entity from the plurality of candidate entities that has a fitness value exceeds a predetermined threshold (Figs. 1A-1B and 2; lines 8-15 of page 40), creating at least one new candidate entity by creating a variation in the selected at least one candidate entity if the selected at least one candidate does not satisfy the predetermined design requirement or a number of iterations has not reached the predetermined value of the iteration count, including performing one of a reproduction operation, offspring crossover operation, mutation operation, and an architecture altering operation on the at least one selected candidate entity (Figs. 1A-1B and 2; line 21 of page 35

to line 6 of page 36, line 7 of page 39 to line 12 of page 41, lines 8-14 of page 43), and terminating the iterations if the selected at least one candidate satisfies the predetermined design requirement or a number of iterations has reached the predetermined value of the iteration count, wherein at least one of the selected candidate entities is used to design an end-result structure in view of the predetermined design requirement, wherein the end-result structure does not possess key characteristics of the reference structure; and updating the iteration count at the end of each iteration (Figs. 1A-1B and 2; line 17 of page 34 to line 7 of page 35, lines 5-16 of page 42).

Independent claim 23 claims a non-transitory computer-readable storage medium having stored thereon executable code which causes a computer to perform a process, for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure. The process includes: initializing a plurality of candidate entities and an iteration count with a predetermined value by supplying, from an external source, at least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities (Figs. 1A-1B and 2; line 12 of page 34 – line 7 of page 35, line 20 of page 41 – line 4 of page 42), wherein each candidate entity is represented by a tree structure having a plurality of nodes representing a structure of the candidate entity (line 7 of page 36 – line 2 of page 37).

Independent claim 23 further performs iterative genetic programming operations, each iteration including: creating a description of the structure for each of the candidate entities based on its tree structure (Figs. 1A-2B and 2; lines 8-13 of page 35, lines 17-19 of page 42), analyzing behavior and characteristics based on the description of the structure of each candidate entity, including a simulation of the structure (Figs. 1A-1B and 2; lines 3-15 of page 37, lines 20-22 of page 42), comparing each of the plurality of candidate entities with the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity (Figs. 1A-1B and 2; lines 6-12 of page 38, lines

1-3 of page 43), the isomorphism value representing a dissimilarity between the respective candidate entity and the reference structure (lines 4-6 of page 89 and line 1 of page 94 to line 7 of page 95), determining a fitness value for each of the candidate entities based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity (Figs. 1A-1B and 2; lines 8-20 of page 35, lines 13-22 of page 38, lines 4-7 of page 43), selecting at least one candidate entity from the plurality of candidate entities that has a fitness value exceeds a predetermined threshold (Figs. 1A-1B and 2; lines 8-15 of page 40), creating at least one new candidate entity by creating a variation in the selected at least one candidate entity if the selected at least one candidate does not satisfy the predetermined design requirement or a number of iterations has not reached the predetermined value of the iteration count, including performing one of a reproduction operation, offspring crossover operation, mutation operation, and an architecture altering operation on the at least one selected candidate entity (Figs. 1A-1B and 2; line 21 of page 35 to line 6 of page 36, line 7 of page 39 to line 12 of page 41, lines 8-14 of page 43), and terminating the iterations if the selected at least one candidate satisfies the predetermined design requirement or a number of iterations has reached the predetermined value of the iteration count, wherein at least one of the selected candidate entities is used to design an end-result structure in view of the predetermined design requirement, wherein the end-result structure does not possess key characteristics of the reference structure; and updating the iteration count at the end of each iteration (Figs. 1A-1B and 2; line 17 of page 34 to line 7 of page 35, lines 5-16 of page 42).

VI. GROUNDS OF REJECTIONS TO BE REVIEWED ON APPEAL

A. Whether claims 1-6, 13, and 15-23 are patentable under 35 U.S.C. §103(a) over Applicant's admitted prior art (APA) in view of U.S. Patent No. 5,867,397 of Koza et al. ("Koza"), article entitled "An Algorithm for Subgraph Isomorphism" of Ullmann ("Ullmann"), and article entitled "An Overview of Genetic Algorithms: Part 2, Research Topics" of Beasley, et al. ("Beasley").

VII. ARGUMENT

A. <u>Claims 1-6, 13, and 15-23 are patentable over APA, Koza, Ullmann, and Beasley</u>

Claims 1-6, 13, and 15-23 stand or fall together. Claim 1 is the representative claim. As discussed above, Appellant's invention as claimed is directed to a method and apparatus for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure, using genetic programming techniques. Specifically, independent claim 1 recites as follows:

> 1. (Previously Presented) A computer-implemented process for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure, the process comprising:
>
> initializing a plurality of candidate entities and <u>an iteration count with a predetermined value by supplying, from an external source, at least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities</u>, wherein each candidate entity is represented by a tree structure having a plurality of nodes representing a structure of the candidate entity;

performing iterative genetic programming operations, each iteration including:

<u>creating a description of the structure for each of the candidate entities based on its tree structure,</u>

<u>analyzing behavior and characteristics based on the description of the structure of each candidate entity, including a simulation of the structure,</u>

<u>comparing each of the plurality of candidate entities with the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity, the isomorphism value representing a dissimilarity between the respective candidate entity and the reference structure</u>,

determining a fitness value for each of the candidate entities <u>based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity</u>,

selecting at least one candidate entity from the plurality of candidate entities that has a fitness value exceeds a predetermined threshold,

creating at least one new candidate entity by creating a variation in the selected at least one candidate entity if the selected at least one candidate does not satisfy the predetermined design requirement or a number of iterations has not reached the predetermined value of the iteration count, including performing one of a reproduction operation, offspring crossover operation, mutation operation, and an architecture altering operation on the at least one selected candidate entity, and

terminating the iterations if the selected at least one candidate satisfies the predetermined design requirement or a number of iterations has reached the predetermined value of the iteration count, wherein at least one of the selected candidate entities is used to design an end-result structure in view of the predetermined design requirement, wherein the end-result structure does not possess key characteristics of the reference structure; and

updating the iteration count at the end of each iteration.

(Emphasis added)

Independent claim 1 is related to a process for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure, which is absent from the alleged APA, Koza, Ullmann, and Beasley. The

Examiner contends that lines 1 and 11-12 of page 9 of the alleged APA discloses such a limitation (1/12/2011 Office Action, page 3). Appellant respectfully disagrees.

Although the cited section of the alleged APA discusses that iterative genetic programming techniques in general can be utilized to create a structure, however, the cited section of the alleged APA fails to specifically disclose that the genetic programming techniques can be utilized to create a structure that can satisfy a design requirement in which at least one characteristic is not found in a reference structure. There is no disclosure or suggestion within the alleged APA for such a limitation.

Although the Examiner acknowledges that the alleged APA fails to disclose the above limitation, the Examiner contends that such a limitation would be obvious to an ordinary skill in the art (1/12/2011 Office Action, pages 7-10). Appellant respectfully disagrees.

In order to render a claim obvious, each and every limitations of the claim must be taught by the cited references. As discussed above, there is no disclosure or suggestion of the above limitation within the alleged APA, Koza, Ullmann, and Beasley. The Examiner mainly relies on the alleged obviousness from one with ordinary skill in the art without providing concrete evidence and thus fails to establish a prima facie case of the obviousness rejection.

In addition, independent claim 1 requires initializing candidate entities and an iteration count with a predetermined value by supplying, from an external source, where at least one candidate entity partially satisfies the predetermined design requirement which includes a characteristic of the reference structure to the initialized candidate entities. It is respectfully submitted that such a limitation is also absent from the combination of alleged APA, Koza, Ullmann, and Beasley.

The Examiner contends that lines 9-10 of page 8 and lines 3-5 of page 9 of the alleged APA disclose such a limitation (Office Action, pages 3-4). Appellant respectfully disagrees.

It is submitted that the cited section of the alleged APA only discloses randomly initializing candidate entities. The cited section of the alleged APA fails to disclose or suggest initializing candidate entities in which at least one candidate entity partially satisfies the predetermined design requirement which includes a characteristic of the reference structure to the initialized candidate entities.

As explained in the specification of the present application, "a human designer may find it advantageous to apply his or her domain knowledge of a particular field to create the initial circuit for a particular problem. Such knowledge would bias the search for a satisfactory design in the direction of a particular known or desirable characteristic." See e.g., page 53, lines 11-16 of specification of the present application ("Specification'). An initial entity "is often a seeded (primed) individual that is believed to be reasonably good." See e.g., page 42, lines 3-4 of the Specification.

The initial random population in the alleged APA does not include "at least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities." There is no suggestion in the alleged APA to include characteristics of the reference structure during initialization. As such, the process of the present application is more efficient over the alleged APA because less iterations would be needed to derive a final candidate since the initialization already bias the process in a direction.

Further, independent claim 1 requires creating a description of the structure for each of candidate entities based on its tree structure and analyzing behavior and characteristics based on the description of the structure. It is respectfully submitted that such a limitation is also absent from the alleged APA, Koza, Ullmann, and Beasley.

The Examiner contends that line 20 of page 6 and lines 3-5 of page 9 disclose the above limitation (1/12/2011 Office Action, pages 4-5). Appellant respectfully disagrees. The cited section of the alleged APA only discloses that genetic programming can be performed based on a tree structure. It is submitted that cited section of the alleged APA fails to disclose creating a description from the structure and analyzing behavior and characteristics based on the description.

Furthermore, independent claim 1 requires comparing each of the candidate entities with the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity, where the isomorphism value represents a dissimilarity between the respective candidate entity and the reference structure, and determining a fitness value for each of the candidate entities based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity. It is respectfully submitted that such a limitation is also absent from the alleged APA, Koza, Ullmann, and Beasley.

Specifically, the cited references fail to disclose deriving or suggesting an isomorphism value for each candidate entity based on the analysis of the behavior and characteristics and determining a fitness value based on at least the isomorphism value using genetic programming techniques.

Although the Examiner acknowledges that Koza fails to disclose the above limitation, the Examiner contends that Ullmann discloses such a limitation (1/12/2011 Office Action, page 9). Appellant respectfully disagrees.

Although Ullmann discloses a method using an isomorphism value, Ullmann does not disclose or suggest such a method can be used in genetic programming to design an entity structure. There is no suggestion within Ullmann to combine with Koza, or vice versa. In fact, there is no mention of Ullmann or the term of "isomorphism" in Koza. The fact that the present application references Ullmann and its isomorphism algorithm in the specification (see Specification, page 94) does not provide any motivation for one with ordinary skill in the art, based on the teachings of Ullmann and Koza, to combine Ullmann and Koza.

In contrast, the only suggestion to utilize an isomorphism algorithm disclosed by Ullmann with the genetic programming techniques can be found on page 94 of the Specification of the present application. Therefore, the Examiner can only combine these two references based on the specification of the present application (e.g., based on the impermissible hindsight of Applicant's own disclosure).

Clearly, the Examiner fails to establish a prima facie case to combine the Koza with Ullmann, because there is absolutely nothing within both references to suggest a combination of both references. The Examiner can only assert such a combination based on the disclosure of Applicant's own disclosure. Therefore, it is respectfully submitted that, without Applicant's own disclosure, it is not obvious to one with ordinary skill in the art to combine these two references.

Even if the Koza were combined with Ullmann, such a combination still lacks the limitations set forth above, particularly, using an isomorphism value to represent

dissimilarity between two entities during iterative operations using genetic programming techniques. Again, any suggestion for combining the Koza and Ullmann can only be found based on the impermissible hindsight of Applicant's own disclosure.

Although Beasley discloses an introduction of genetic algorithms, Beasley fails to disclose the specific limitations of genetic programming techniques set forth in independent claim 1. Independent claim 1 is not merely about genetic algorithms or genetic programming. Rather, independent claim 1 is about a specific way to design structures using genetic programming techniques.

Again, in order to render a claim obvious, each and every limitations of the claim must be taught by the cited references, individually or in combination. It is respectfully submitted that Koza, Ullmann, and Beasley, individually or in combination, fail to disclose or suggest each and every limitations of independent claim 1. Therefore, for reasons set forth above, it is respectfully submitted that independent claim 1 is patentable over the alleged APA, Koza, Ullmann, and Beasley.

Similarly, independent claims 22-23 include limitations similar to those recited in claim 1. Thus, for the reasons similar to those discussed above, independent claims 22-23 are patentable over the alleged APA, Koza, Ullmann, and Beasley. Given that the rest of the claims depend from one of the above independent claims, at least for the reasons similar to those discussed above, it is respectfully submitted that the rest of the claims are patentable over the alleged APA, Koza, Ullmann, and Beasley.

VIII. CONCLUSION

For the reasons stated above, claims 1-6, 13, and 15-23 are patentable under 35 U.S.C. § 103(a) over the alleged APA, Koza, Ullmann, and Beasley. Appellant respectfully requests that the Board reverse the rejections of the claims 1-6, 13, and 15-23 and direct the Examiner to enter a Notice of Allowance for claims 1-6, 13, and 15-23.

Enclosed is the amount to cover the fee for filing a brief in support of an appeal as required under 37 C.F.R. § 1.17(c) and 41.20(b)(2).

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due. Furthermore, if an extension is required, then Appellant hereby requests such extension.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: July 12, 2011

/Kevin G. Shao/

Kevin G. Shao
Attorney for Appellant
Registration No. 45,095

Customer No. 008791
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300

APPENDIX A: Claims on Appeal

The claims on appeal read as follows:

1. (Previously Presented) A computer-implemented process for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure, the process comprising:

initializing a plurality of candidate entities and an iteration count with a predetermined value by supplying, from an external source, at least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities, wherein each candidate entity is represented by a tree structure having a plurality of nodes representing a structure of the candidate entity;

performing iterative genetic programming operations, each iteration including:

creating a description of the structure for each of the candidate entities based on its tree structure,

analyzing behavior and characteristics based on the description of the structure of each candidate entity, including a simulation of the structure,

comparing each of the plurality of candidate entities with the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity, the isomorphism value representing a dissimilarity between the respective candidate entity and the reference structure,

determining a fitness value for each of the candidate entities based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity,

selecting at least one candidate entity from the plurality of candidate entities that has a fitness value exceeds a predetermined threshold,

creating at least one new candidate entity by creating a variation in the selected at least one candidate entity if the selected at least one candidate does not satisfy the predetermined design requirement or a number of iterations has not reached the predetermined value of the iteration count, including performing one of a reproduction operation, offspring crossover operation, mutation operation, and an architecture altering operation on the at least one selected candidate entity, and

terminating the iterations if the selected at least one candidate satisfies the predetermined design requirement or a number of iterations has reached the predetermined value of the iteration count, wherein at least one of the selected candidate entities is used to design an end-result structure in view of the predetermined design requirement, wherein the end-result structure does not possess key characteristics of the reference structure; and

updating the iteration count at the end of each iteration.

2. (Original) The process defined in Claim 1 wherein creating at least one new candidate entity comprises mutating the at least one candidate entity.

3. (Original) The process defined in Claim 2 wherein selecting the at least one candidate entity is performed by simulating annealing.

4. (Previously Presented) The process defined in Claim 2 wherein selecting at least one candidate is performed by hill climbing.

5. (Original) The process defined in Claim 1 wherein the at least one candidate entity is a member of a population of entities.

6. (Original) The process defined in Claim 5 wherein creating at least one new candidate entity comprises performing a crossover operation among a group of candidate entities, the group of entities comprising the selected entity and at least one other entity from the population, the at least one new coordinate entity created by crossover comprising at least a portion of the selected entity and at least a portion of that at least one other entity.

7. – 12. (Cancelled)

13. (Original) The process defined in Claim 1 further comprising creating the at least one candidate entity by a random process.

14. (Cancelled)

15. (Previously Presented) The process defined in Claim 1 further comprising: evaluating the selected at least one candidate entity by simulating the selected at least one candidate entity to ascertain whether the selected at least one candidate entity more closely satisfies the design requirement.

16. (Previously Presented) The process defined in Claim 1 further comprising: evaluating the selected at least one candidate entity by observing a physical realization representing the selected at least one candidate entity to ascertain whether the selected at least one candidate entity more closely satisfies the design requirement.

17. (Original) The process defined in Claim 1 wherein the candidate entity conforms to a constrained syntactic structure.

18. (Original) The process defined in Claim 1 wherein the candidate entity comprises an electrical circuit.

19. (Original) The process defined in Claim 1 wherein the candidate entity comprises a controller.

20. (Original) The process defined in Claim 1 wherein the candidate entity comprises an antenna.

21. (Original) The process defined in Claim 1 wherein the candidate entity comprises a mechanical system.

22. (Previously Presented) A computer system, comprising:

a processor; and

a memory coupled to the processor for storing computer executable instructions, which when executed from the memory, cause the processor to perform a process for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure, the process comprising:

initializing a plurality of candidate entities and an iteration count with a predetermined value by supplying, from an external source, at least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities, wherein each candidate entity is represented by a tree structure having a plurality of nodes representing a structure of the candidate entity;

performing iterative genetic programming operations, each iteration including:

creating a description of the structure for each of the candidate entities based on its tree structure,

analyzing behavior and characteristics based on the description of the structure of each candidate entity, including a simulation of the structure,

comparing each of the plurality of candidate entities with the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity, the isomorphism value representing a dissimilarity between the respective candidate entity and the reference structure,

determining a fitness value for each of the candidate entities based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity,

selecting at least one candidate entity from the plurality of candidate entities that has a fitness value exceeds a predetermined threshold,

creating at least one new candidate entity by creating a variation in the selected at least one candidate entity if the selected at least one candidate does not satisfy the predetermined design requirement or a number of iterations has not reached the predetermined value of the iteration count, including performing one of a reproduction operation, offspring crossover operation, mutation operation, and an architecture altering operation on the at least one selected candidate entity, and

terminating the iterations if the selected at least one candidate satisfies the predetermined design requirement or a number of iterations has reached the predetermined value of the iteration count, wherein at least one of the selected candidate entities is used to design an end-result structure in view of the predetermined design requirement, wherein the end-result structure does not possess key characteristics of the reference structure; and

updating the iteration count at the end of each iteration.

23. (Previously Presented) A non-transitory computer-readable storage medium having stored thereon executable code which causes a computer to perform a process, for creating an entity that satisfies a predetermined design requirement that at least one characteristic is not in a reference structure, the process comprising:

initializing a plurality of candidate entities and an iteration count with a predetermined value by supplying, from an external source, at least one candidate entity partially satisfying the predetermined design requirement which includes a characteristic of the reference structure to the initialized plurality of candidate entities, wherein each candidate entity is represented by a tree structure having a plurality of nodes representing a structure of the candidate entity;

performing iterative genetic programming operations, each iteration including:

creating a description of the structure for each of the candidate entities based on its tree structure,

analyzing behavior and characteristics based on the description of the structure of each candidate entity, including a simulation of the structure,

comparing each of the plurality of candidate entities with the reference structure based on the analysis of the behavior and characteristics to obtain an isomorphism value for each candidate entity, the isomorphism value representing a dissimilarity between the respective candidate entity and the reference structure,

determining a fitness value for each of the candidate entities based on a compliance with the predetermined design requirement and the isomorphism value of the respective candidate entity,

selecting at least one candidate entity from the plurality of candidate entities that has a fitness value exceeds a predetermined threshold,

creating at least one new candidate entity by creating a variation in the selected at least one candidate entity if the selected at least one candidate does not satisfy the predetermined design requirement or a number of iterations has not reached the predetermined value of the iteration count, including performing one of a reproduction operation, offspring crossover operation, mutation operation, and an architecture altering operation on the at least one selected candidate entity, and

terminating the iterations if the selected at least one candidate satisfies the predetermined design requirement or a number of iterations has reached the predetermined value of the iteration count, wherein at least one of the selected candidate entities is used to design an end-result structure in view of the predetermined design requirement, wherein the end-result structure does not possess key characteristics of the reference structure; and

updating the iteration count at the end of each iteration.

APPENDIX B: Evidence

None.

APPENDIX C: Related Proceedings

None.